Overview: Translation stages

- ./julia (repl.c)

- eval/include (builtins.c)
  –> parser (julia-parser.scm)
    –> run macros (ast.c)
      –> lowered form (julia-syntax.scm)
        –> execute toplevel expressions (toplevel.c)
          –> call functions (gf.c)
            –> dispatch (typemap.c)
              –> interpreter (interpreter.c) OR
                 compiler (codegen.cpp)
                   + optimizations (inference.jl)

For details, see Jeff's talk from JuliaCon 2014.

Overview: Standard Library

- ./julia (repl.c)
  –> getopt parsing
  –> loads the standard library

- libjulia.so (or .dll or .dylib)
  –> Garbage Collector / Allocator
  –> Builtin Functions / Intrinsics
  –> Interpreter
  –> Builtin Types
  –> Type System
  –> Dispatch System
  –> Compiler
  –> Platform abstraction / integration
  –> System Image / serializer

- sys.so (and other lib/julia/ files *.so and *.ji)
  –> Core module
  –> Inference module
  –> Base module

- Packages

- Dependencies
  –> LLVM, BLAS, libgit2, etc.